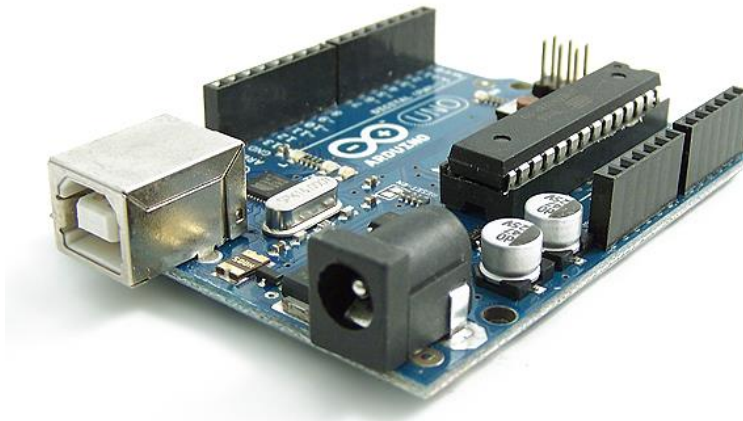


4CS016 Embedded Systems Programming Workbook 2



Contents

4CS016 Embedded Systems Programming Workbook 2	1
Introduction	3
Lab 1. Getting Started with the Arduino.....	4
Complete Activity 2.1 of the Portfolio	7
Lab 2: Error finding.....	8
Lab 3. Prototyping with the breadboard.	11
Complete Activity 2.2 of the Portfolio	13
Complete Activity 2.3 of the Portfolio	13

Introduction

This workbook complements the lectures for 4CS016 as such, the lecture notes will be referred to as the work progresses. You will also need to complete portfolio activities as you progress, these are highlighted as you go along.

This workbook is intended to be completed during teaching week 3 and 4

There are activities that need to be completed and put into the Portfolio on CANVAS, in the assignment folder.

Note on using TinkerCAD

You don't need to set up COM ports on TinkerCAD, but can go straight to coding.

Lab 1. Getting Started with the Arduino

Step 1: Open the Arduino programming software Sketch.

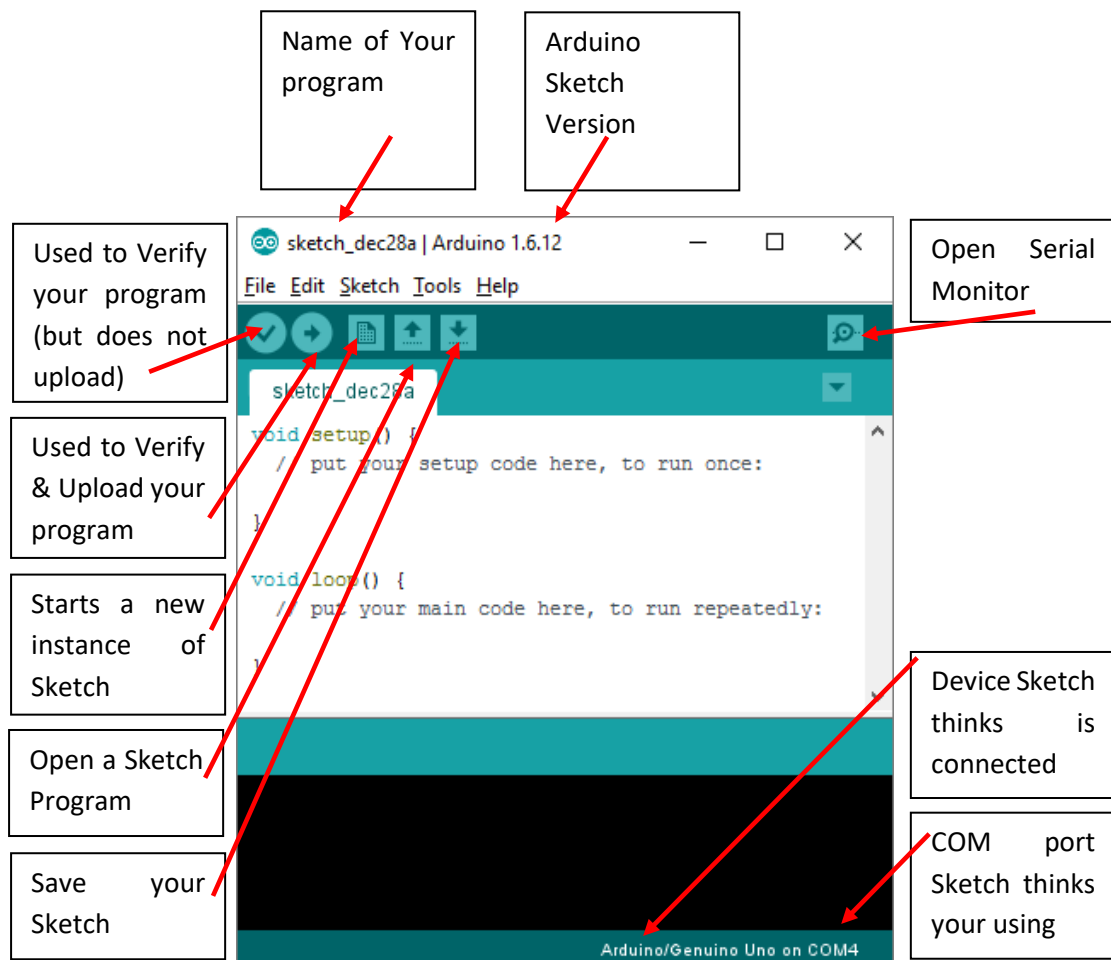


Figure 1: Sketch Overview

Step 2: Connect the Arduino in the kit to the USB port on the PC.

Step 3: Check to ensure the Arduino Uno is selected (see below)

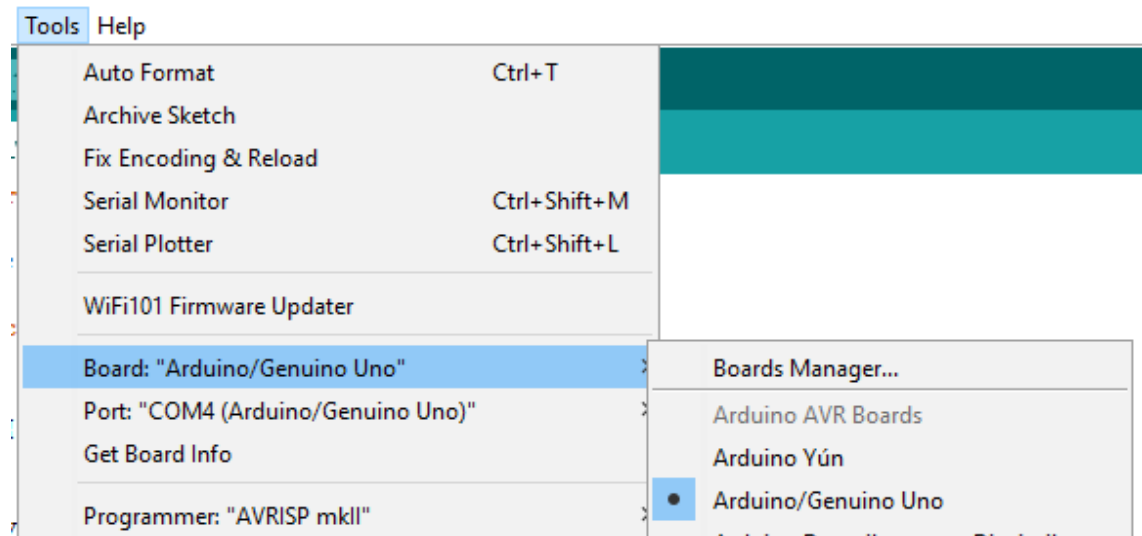


Figure 2: Board Selection

Step 4: Check if the correct COM port is showing (see below)

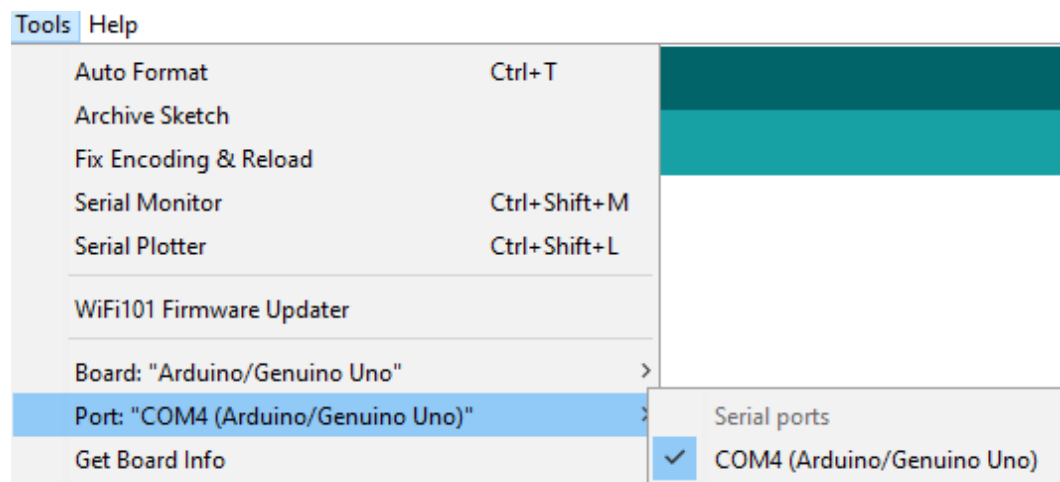


Figure 3: COM port selection

Don't worry if you see a different Serial Port here (Com port).

Step 5: Check that the power light is illuminated.

If all the above steps are complete, you are ready to start programming 😊

If not speak to your workshop Supervisor.

Step 6: Open the Example Blink shown in Lecture 2. This can be achieved by selecting

- File
- Examples
- 01.Basic
- Blink



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
  of use the correct LED pin whatever is the board used.
  If you want to know what pin the on-board LED is connected to on your Arduino model,
  the Technical Specs of your board at https://www.arduino.cc/en/Main/Products

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald

  modified 2 Sep 2016
  by Arturo Guadalupi
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

Done uploading.

avrdude done. Thank you.

Arduino/Genuino Uno on COM4

Figure 4: Blink Example

Note: An explanation of the code is given in Lecture 2, so read the slides to understand the program in detail, but basically it flashes the on-board LED once every second. Remember the on-board LED is connected to pin 13

Step 7: Upload the program to your board, you should know this has been successful for several reasons (on TinkerCAD just execute the circuit)

- 1) The LED on the board flashes every 1 second.
- 2) The Sketch program thinks its successfully sent the program i.e. Says Done uploading and no error as shown below.

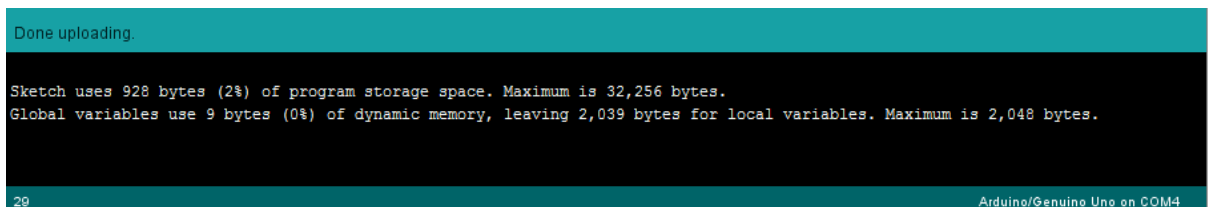


Figure 5: Successful Upload panel.

If your LED does not flash once a second, ask your workshop supervisor for help.

Step 8: Adjust your program so that

- the LED displays the binary for the decimal number 63
- then pauses off for 5 seconds
- before repeating this.
- Use LED on for a 1
- Use LED off for a zero
- Each digit of binary should be 1 second.

Complete Activity 2.1 of the Portfolio

End of Lab 1

Please continue with Lab 2

Lab 2: Error finding

Step 1: We are going to introduce an error into the program and observe the results. It doesn't really matter where the error is, but let's change the following

```
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);                    // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);                    // wait for a second  
}
```

To

```
void loop() {  
  digitalwrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);                    // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);                    // wait for a second  
}
```

So we have changed the upper case "W" to a lower case. This is a common issue and can occasionally be hard to spot. Try to Upload your program.

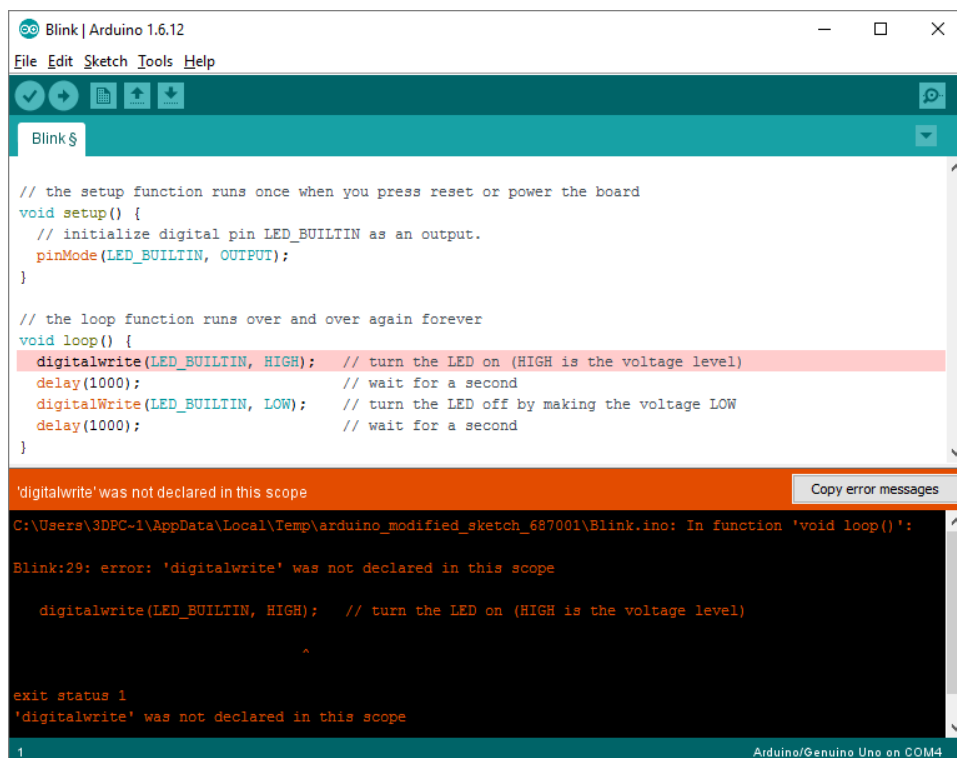


Figure 6: Typical program compilation error

The Sketch IDE is quite good at helping us to debug so here's why

```
C:\Users\3DPC-1\AppData\Local\Temp\arduino_modified_sketch_687001\Blink.ino: In function 'void loop()':
```

This tells us there's is an error in the function loop();

```
Blink:29: error: 'digitalwrite' was not declared in this scope
```

Blink is our program

29 is the line the error is found on

"digitalwrite" is highlighted as a variable that has not been declared, this is because "digitalWrite" is a method that exists, we know this, it appears orange/brown, but "digitalwrite" appears black.

```
digitalwrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
```

Shows us the actual line of code causing issues.

```
exit status 1
```

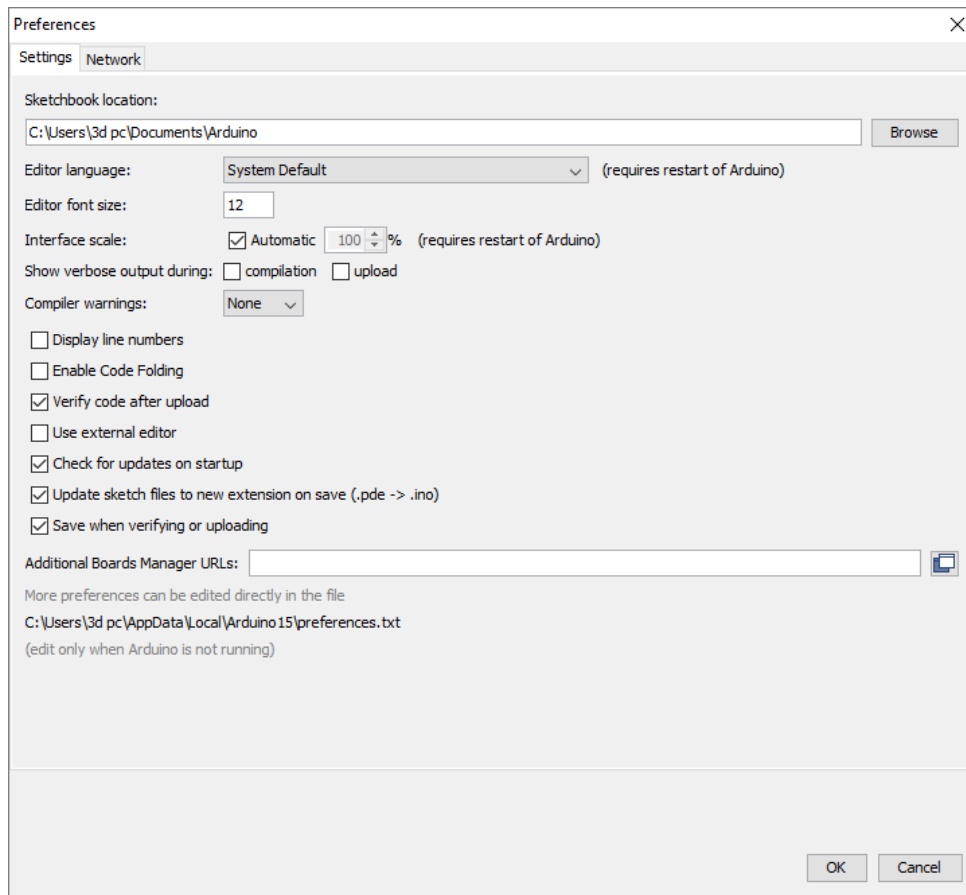
An exit status of 1 means that there was a problem compiling, linking, or uploading some code.
(Which one, we shall get to shortly)

```
'digitalwrite' was not declared in this scope
```

This final line tried to be helpful and inform us we haven't declared the variable "digitalwrite" locally (or globally).

Step2: The Arduino IDE contains a more detailed debugging mode, so let's switch that on.

Step 3: Click File, then Preferences.



Step 4: Tick "Show verbose output during", "Compilation" & "Upload".

Step 5: Click OK.

Step 6: Remove the error from your program and recompile it.

End of Lab 2

Please continue with Lab 3

Lab 3. Prototyping with the breadboard.

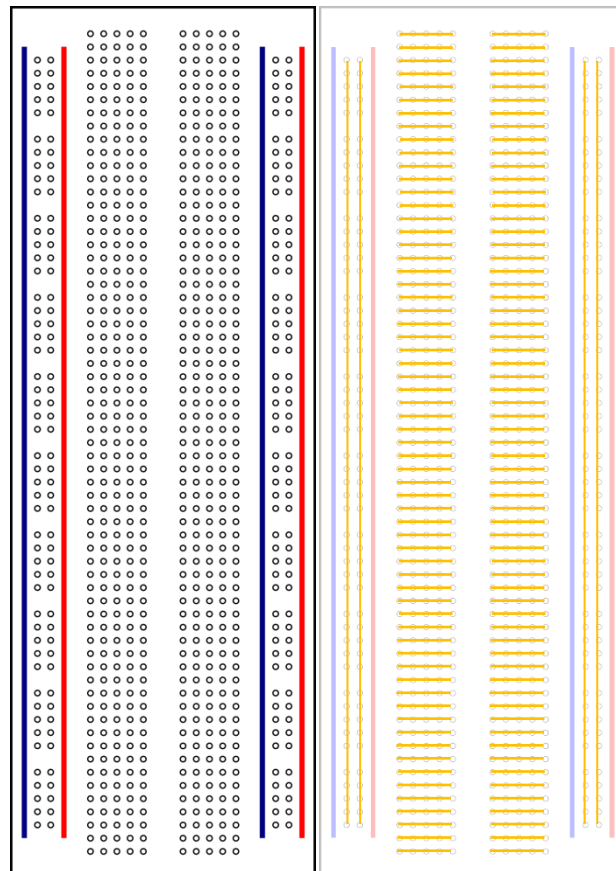


Fig 7: Breadboard on the left, underlying wire connections between the holes on the right

In the figure above, we can see that anything on the same horizontal line in the central two groups of holes will be connected to each other (as if they'd been soldered together) and anything plugged in on the same vertical line in the two outer groups of holes will be connected together.

Step 1. Wire up the Circuit from the Lecture

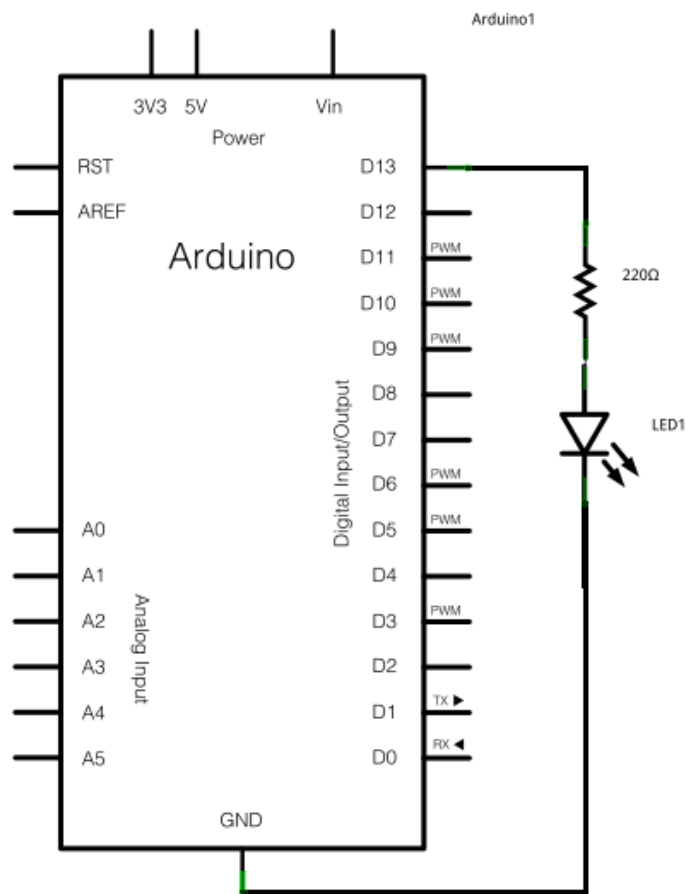


Figure 8. Blinking LED circuit

Step 2: Change the code so that the LED flashes based on a variable called `ledOne` instead of `"LED_BUILTIN"`.

Step 3: Change the code so that the LED flashes from Pin 12 instead of 13.

Step 4: we are going to create a 4 bit LED display that counts from 0 to 15 using the LED's

- You will need to use 4 LED's
- Create 3 more variables `ledTwo`, `ledThree`, and `ledFour`.
- The pattern should look like this.

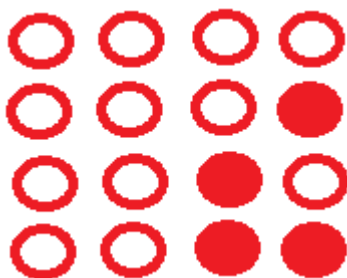


Figure 9. Led Display example

My example in figure 9 goes up to 3, we want you to continue to 15.

Complete Activity 2.2 of the Portfolio

Step 5: Create a set of traffic lights using a DO while loop for 5 iterations, it should then stop, sequence is

- Red
- Red + Amber
- Green
- Then
- Amber
- Repeat.

Complete Activity 2.3 of the Portfolio

End of Lab 3

End of Workbook 2